Faster and Deterministic Subtrajectory Clustering

Ivor van der Hoog[1], Thijs van der Horst[2], and Tim Ophelders[2]

1 **Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark**
vanderhoog@gmail.com

2 **Department of Information and Computing Sciences, Utrecht University, the Netherlands**
Department of Mathematics and Computer Science, TU Eindhoven, the Netherlands
{t.w.j.vanderhorst|t.a.e.ophelders}@uu.nl

Abstract

We study the subtrajectory clustering problem. Given a trajectory T, the goal is to identify a set of subtrajectories such that each point on T is included in at least one subtrajectory, and subsequently group these subtrajectories together based on similarity under the Fréchet distance. We wish to minimize the set of groups. This problem was shown to be NP-complete by Akitaya, Brüning, Chambers, and Driemel (2021), and the focus has mainly been on approximation algorithms. We study a restricted variant, where we may only pick subtrajectories that start and end at vertices of T, and give an approximation algorithm that significantly improves previous algorithms in both running time and space, whilst being deterministic.

Related Version Full version: arXiv:2402.13117

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 899987. Tim Ophelders is partially supported by the Dutch Research Council (NWO) under the project number VI.Veni.212.260.

1 Introduction

Buchin *et al.* [8] proposed the subtrajectory clustering problem. The goal is to partition an input trajectory T of n vertices into subtrajectories, and to group these subtrajectories into *clusters* such that all subtrajectories in a cluster have low Fréchet distance to one another. The clustering under the Fréchet distance is a natural application of Fréchet distance and a well-studied topic [9, 10, 11, 14, 15] that has applications in for example map reconstruction [6, 7]. Throughout recent years, several variants of the algorithmic problem have been proposed [1, 3, 5, 8]. Agarwal *et al.* [1] aim to give a general definition for subtrajectory clustering by defining a function f that evaluates the quality of a set of clusters C. Their definition, however, does not encompass the definition in [8] and has nuances with respect to [3, 5, 13]. Regardless of variant, the subtrajectory clustering problem has been shown to be NP-complete [1, 3, 8]. Agarwal *et al.* [1] therefore propose a bicriterial approximation scheme. They present a heuristic algorithm for the following. Suppose that we are given some $\Delta \geq 0$. Let k denote the smallest integer such that there exists a clustering C with k clusters with score $f(C) \leq \Delta$. The goal is to compute a clustering C' with $\mathcal{O}(k \operatorname{polylog} n)$ clusters and score $f(C) \in \Theta(\Delta)$.

40th European Workshop on Computational Geometry, Ioannina, Greece, March 13–15, 2024.
This is an extended abstract of a presentation given at EuroCG'24. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

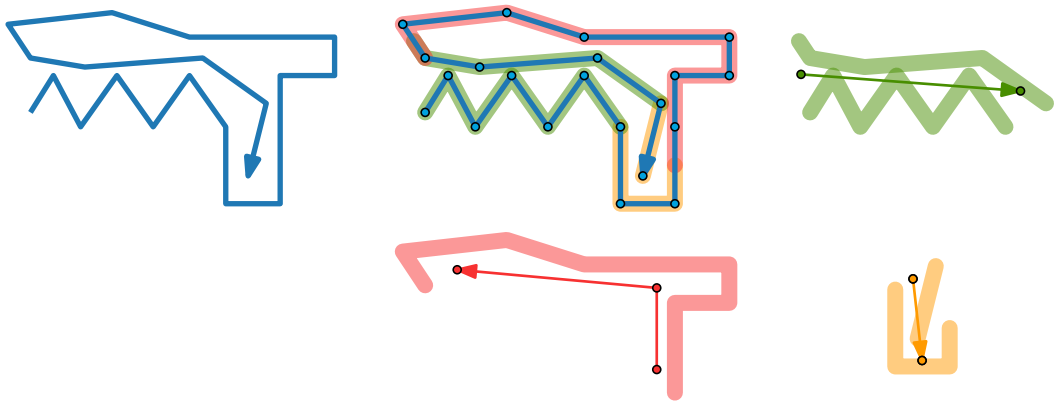

Figure 1 The trajectory T (blue, left) is covered by four pathlets. Each pathlet is given by a reference curve (green, red, yellow) and the subcurve(s) of T the curve covers.

Akitaya *et al.* [3] present a less general, but more computable, bicriterial approximation problem: suppose that we are given some $\Delta \geq 0$ and integer $\ell \geq 1$. An (ℓ, Δ)-clustering is a set C^* of clusters (sets of subtrajectories) where (Figure 1):

- C^* *covers* T: for all points t on T, there is a cluster $Z \in C^*$ and curve $S \in Z$ containing t, and
- every cluster $Z \in C^*$ contains a "reference curve" P_Z with at most ℓ vertices, and
- for every $Z \in C^*$, all subtrajectories $Q \in Z$ have $d_F(P_Z, Q) \leq \Delta$.

Under the discrete Fréchet distance, [3] compute an $(\ell, 19\Delta)$-clustering C of $\mathcal{O}(k\ell^2 \log k\ell)$ size, using $\mathcal{O}(n)$ space and $\tilde{\mathcal{O}}(kn^2)$ expected running time. Brüning *et al.* [5] compute, under the continuous Fréchet distance, an $(\ell, \Theta(\Delta))$-clustering of $\mathcal{O}(k\ell \log k\ell)$ size (where the constant in $\Theta(\Delta)$ is considerably large). Their algorithm uses $\tilde{\mathcal{O}}(n^3)$ space and has $\tilde{\mathcal{O}}(kn^3)$ expected running time. Recently, Conradi and Driemel [13] improve both the size and the distance of the clustering. Under the continuous Fréchet distance, they compute an $(\ell, 11\Delta)$-clustering of $\mathcal{O}(k \log n)$ size using $\tilde{\mathcal{O}}(n^4\ell)$ space and $\tilde{\mathcal{O}}(kn^4\ell + n^4\ell^2)$ time.

Contribution. In this abstract, we give significant improvements in both space and running time complexities, for the restricted case where the clustering needs to be *target-discrete*. A clustering is target-discrete whenever all subtrajectories in a cluster are restricted to have their endpoints lie on vertices of the input. Under this natural restriction, even under the continuous Fréchet distance, our algorithm is near-quadratic and uses near-quadratic space. In the full version we further lower the space used to $\mathcal{O}(n\ell \log n)$. Additionally, in the full version we investigate the unrestricted setting, as well as other variants.

2 Problem Statement

Curves and subcurves. A *curve* (or polyline) with n vertices is a piecewise-linear map $P\colon [1, n] \to \mathbb{R}^d$ whose breakpoints (called *vertices*) are at each integer parameter, and whose pieces are called *edges*. For $1 \leq a \leq b \leq n$, we define the *subcurve* $P[a, b]$ of P that starts at $P(a)$ and ends at $P(b)$. If a and b are integers, we call $P[a, b]$ a *vertex-subcurve* of P.

Fréchet distance. We define the Fréchet distance using the concept of the *free space diagram*. For $\Delta \geq 0$, the Δ-free space diagram is defined as follows.

▶ **Definition 2.1** (Free space diagram)**.** For two curves P and Q with n and m vertices, respectively, the Δ-*free space diagram* Δ-FSD(P,Q) of P and Q is the set of all points $(x,y) \in [1,n] \times [1,m]$ in their parameter space with Euclidean distance $d(P(x),Q(y)) \leq \Delta$.

The *grid cells* of the free space diagram are the squares $[i,i+1] \times [j,j+1]$ for integers i and j. The *obstacle space* of Δ-FSD(P,Q) is its complement $([1,n] \times [1,m]) \setminus \Delta$-FSD$(P,Q)$.

Alt and Godau [4] observed that two curves P and Q have Fréchet distance at most Δ between them precisely if there exists a bimonotone path from $(1,1)$ to (n,m) in Δ-FSD(P,Q).

Input and output. We consider clustering a curve T with n vertices that we call the *trajectory*. Given parameters $\ell \in \mathbb{N}$ and $\Delta \geq 0$, we consider constructing an (ℓ,Δ)-*clustering* of T, which is a set of (ℓ,Δ)-*pathlets*:

▶ **Definition 2.2** (Pathlet)**.** An (ℓ,Δ)-*pathlet* is a tuple $(P,\mathcal{I})$ where P is a curve with at most ℓ vertices and $\mathcal{I}$ is a set of intervals in $[1,n]$ where $d_F(P,T[a,b]) \leq \Delta$ for all $[a,b] \in \mathcal{I}$.

We call P the *reference curve* of $(P,\mathcal{I})$. We call the pathlet *target-discrete* if all intervals in $\mathcal{I}$ have integer endpoints. The *coverage* of a pathlet is $\text{Cov}(P,\mathcal{I}) = \bigcup \mathcal{I}$.

We can see a pathlet $(P,\mathcal{I})$ as a cluster, where the center is P and all subtrajectories induced by $\mathcal{I}$ get mapped to P. The coverage of a set of pathlets C is $\text{Cov}(C) := \bigcup_{(P,\mathcal{I}) \in C} \text{Cov}(P,\mathcal{I})$. An (ℓ,Δ)-*clustering* is a set C of (ℓ,Δ)-pathlets with $\text{Cov}(C) = [1,n]$.

3 Quality of greedy algorithm

Subtrajectory clustering is closely related to the *set cover* problem. In this problem, we have a discrete universe $\mathcal{U}$ and a family of sets $\mathcal{S}$ in this universe, and the goal is to pick a minimum number of sets in $\mathcal{S}$ such that their union is the whole universe. For subtrajectory clustering with target-discrete pathlets, we can set $\mathcal{U} = \{[i,i+1] \mid i \in [1,n-1] \cap \mathbb{N}\}$ and $\mathcal{S} = \{[i,i+1] \in \bigcup \mathcal{I} \mid \text{there exists an } (\ell,\Delta)\text{-pathlet } (P,\mathcal{I})\}$. That is, the universe is the set of intervals parameterizing the edges of T, and $\mathcal{S}$ is the family of sets of edge parameterizations of T that can be covered by a pathlet.

The decision variant of set cover is NP-complete [17]. However, the following greedy strategy gives an $\mathcal{O}(\log n)$ approximation of the minimal set cover size [12]. Suppose we have picked a set $\hat{\mathcal{S}} \subseteq \mathcal{S}$ that does not yet cover all of $\mathcal{U}$. The idea is then to add a set $S \in \mathcal{S}$ that maximizes $S \cap (\mathcal{U} \setminus \hat{\mathcal{S}})$, and repeat the procedure until $\mathcal{U}$ is fully covered.

We use this greedy strategy to focus on constructing a pathlet that covers the most uncovered edges of T. In other words, we greedily grow a set of pathlets C, each time adding a (ℓ,Δ)-pathlet $(P,\mathcal{I})$ that (approximately) maximizes the coverage $\|\text{Cov}(P,\mathcal{I}) \setminus \text{Cov}(C)\|$.

4 Pathlet-preserving simplification

Naively, the subtrajectory clustering problem searches over an infinite number of reference curves for the pathlets, namely, all curves in $\mathbb{R}^d$. In prior work, Brüning *et al.* [5] used a simplification of T that significantly reduces the search space at the cost of increasing Δ by a factor 11, with the assumption that reference curves should be line segments. We propose an alternative solution that increases Δ by only a factor 4, while also giving guarantees for reference curves of arbitrary length.

Our simplification algorithm combines the simplification algorithms of Guibas *et al.* [16] and Agarwal *et al.* [2]. The simplification is essentially a greedy procedure. Consider moving along T, starting at $T(1)$. We look for a locally maximal value t for which $d_F(T[1,t], \overline{T(1)T(t)}) \leq \Delta$ (see fig. 2), and we replace $T[1,t]$ by the directed line segment $\overline{T(1)T(t)}$. We then recursively apply this procedure to the subcurve $T[t,n]$, until $t = n$.

Let S be the resulting curve. In the full version we prove Lemma 4.1.

▶ **Lemma 4.1.** *The curve S has Fréchet distance at most Δ to T, and has the property that for any (ℓ, Δ)-pathlet $(P, \mathcal{I})$, there exists an $(\ell + 2, 4\Delta)$-pathlet $(P', \mathcal{I})$ with the same coverage, where P' is a subcurve of S. Moreover, we can construct S in $\mathcal{O}(n \log n)$ time.*

The consequence of simplification. The above result lets us focus on subcurves of S as reference curves only (see Figure 3). Still, the endpoints of reference curves may lie anywhere on S. However, any such reference curve $S[a,b]$ is either a segment (if $\lfloor a \rfloor = \lfloor b \rfloor$), or it can be decomposed into three subcurves: $S[a, \lceil a \rceil]$, $S[\lceil a \rceil, \lfloor b \rfloor]$ and $S[\lfloor b \rfloor, b]$, the second of which is a vertex-subcurve, and the other two are a prefix and suffix of an edge, respectively. This observation leads to the following algorithm (Algorithm 1) where we iteratively grow a set of (ℓ, Δ')-pathlets C (where $\Delta' \in \Theta(\Delta)$). In essence, we run the greedy algorithm of Section 3. Each iteration, let C be the current set of pathlets and let $(P, \mathcal{I})$ be the (ℓ, Δ)-pathlet that maximizes the coverage $\|\mathrm{Cov}(P, \mathcal{I}) \backslash \mathrm{Cov}(C)\|$. We compute:

- a vertex-subcurve with maximal coverage, and
- a prefix or suffix of an edge e with maximal coverage.

Via our above argument, the coverage $\|\mathrm{Cov}(P', \mathcal{I}') \backslash \mathrm{Cov}(C)\|$ of the pathlet $(P', \mathcal{I}')$ that we compute with maximal coverage is at least $\frac{1}{3}$'rd of $\|\mathrm{Cov}(P, \mathcal{I}) \backslash \mathrm{Cov}(C)\|$.

In this abstract, we restrict ourselves to vertex-subcurves of S, constructing a *vertex-to-vertex* (ℓ, Δ')-pathlet $(P, \mathcal{I})$, where P is a vertex-subcurve of S and $\Delta' = 4\Delta$. In the full version, we show how to construct an optimal $(2, \Delta')$-pathlet $(P, \mathcal{I})$, where P is a prefix or suffix of an edge of S.

5 Constructing pathlets with a given reference curve

Suppose first that we have been given the reference trajectory $S[i, i']$ and need to construct an optimal (ℓ, Δ')-pathlet $(S[i, i'], \mathcal{I})$ using it (for $|S[i, i']| \leq \ell$ and some given Δ'). We then proceed as follows. For every integer $j' \in [1, n]$, we consider the minimum integer j for which $d_F(S[i, i'], T[j, j']) \leq \Delta'$, and if it exists, we create a unique interval $I_{j'} = [j, j']$ and add it to $\mathcal{I}$. This trivially maximizes the coverage $\|\mathrm{Cov}(S[i, i'], \mathcal{I}) \setminus \mathrm{Cov}(C)\|$ for any set of pathlets C, and thus gives an optimal pathlet given $S[i, i']$.

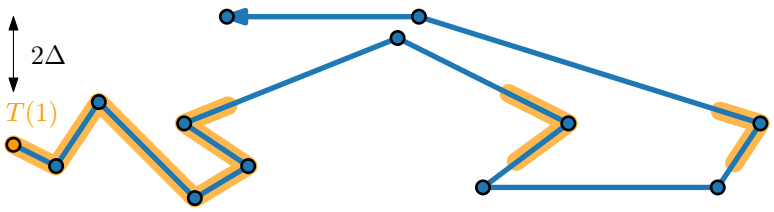

Figure 2 A trajectory T with all points $T(t)$ where $d_F(T[1,t], \overline{T(1)T(t)}) \leq \Delta$ indicated in orange. The last point of each orange connected component may be used for the simplification.

Construct a pathlet-preserving simplification (S, M^{ST}, M^{TS}) of T
Set $\Delta' \leftarrow 4\Delta$
Set $C \leftarrow \emptyset$
while $\mathrm{Cov}(C) \neq [1, n]$ **do**
  Construct a vertex-to-vertex (ℓ, Δ')-pathlet $(P_{\mathrm{ver}}, \mathcal{I}_{\mathrm{ver}})$
  Construct a prefix $(2, \Delta')$-pathlet $(P_{\mathrm{pre}}, \mathcal{I}_{\mathrm{pre}})$
  Construct a suffix $(2, \Delta')$-pathlet $(P_{\mathrm{suf}}, \mathcal{I}_{\mathrm{suf}})$
  Set $(P, \mathcal{I}) \in \{(P_{\mathrm{ver}}, \mathcal{I}_{\mathrm{ver}}), (P_{\mathrm{pre}}, \mathcal{I}_{\mathrm{pre}}), (P_{\mathrm{suf}}, \mathcal{I}_{\mathrm{suf}})\}$ to be the pathlet with maximum coverage over $[1, n] \setminus \mathrm{Cov}(C)$
  Add $(P, \mathcal{I})$ to C
return C

Algorithm 1: SubtrajectoryClustering(T, ℓ, Δ)

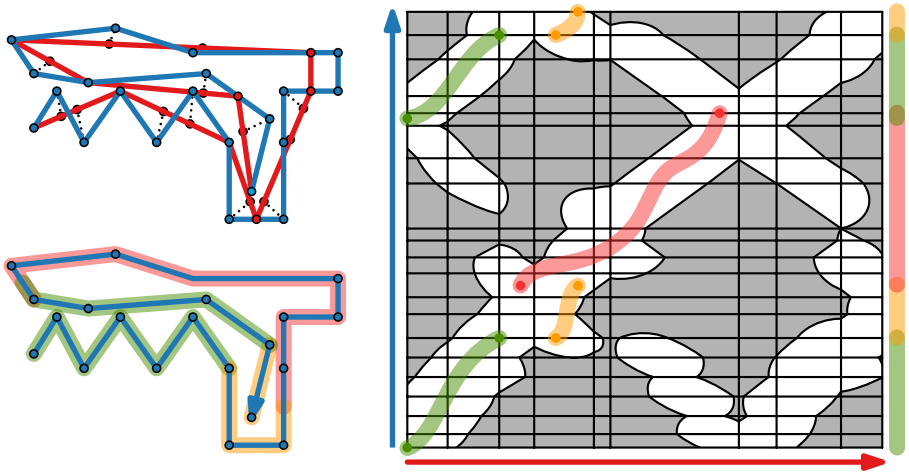

Figure 3 The simplification S of T (top-left, red). The bottom-left clustering of T may be achieved by the right set of paths in Δ'-FSD(S, T). Endpoints of paths with the same color must align vertically, and the horizontal projections of the paths must cover the whole vertical axis.

We compute the above intervals using a data structure that stores information about certain points in the free space diagram. The types of queries we use are:

1. given integers i, i', j and j', determine if $d_F(S[i, i'], T[j, j']) \leq \Delta'$, and
2. given integers i, i' and j', determine if $d_F(S[i, i'], T[j, j']) \leq \Delta'$ for some integer j.

▶ **Lemma 5.1.** *In $\mathcal{O}(n^2 \log n)$ time, we can preprocess S and T into a data structure of $\mathcal{O}(n^2 \log n)$ size, such that queries of type **1** can be answered in $\mathcal{O}(\log n)$ time, and queries of type **2** can be answered in $\mathcal{O}(1)$ time.*

With the above data structure, we compute the intervals $I_{j'}$ in $\mathcal{O}(n \log n)$ time altogether. First, compute the first integer j' for which (i', j') is reachable by some point (i, j) in column i. This takes j' queries of type **2**. All intervals $I_{k'}$ for $k' < j'$ are empty. Next, we compute the first integer j for which $d_F(S[i, i'], T[j, j']) \leq \Delta'$. This takes j queries of type **1**. We then set $I_{j'} = [j, j']$.

Observe that by planarity of the free space diagram, no non-empty interval $I_{k'} = [k, k']$ with $k' \geq j'$ has $k < j$. This is because if $I_{k'} = [k, k']$, then there exists a bimonotone path from (i, k) to (i', k'), which intersects the bimonotone path from (i, j) to (i', j'). Thus there

also exists a bimonotone path from (i, k) to (i', j'), and since $k < j$, we would have that $I_{j'} = [k, j']$.

With the above observation, we can ignore the interval $[1, j-1]$ for the first endpoints of the intervals $I_{k'}$, for $k' > j'$. It follows that we use only $\mathcal{O}(n)$ queries, of either type, to determine all intervals. This gives the following result:

▶ **Theorem 5.2.** *Let C be a set of target-discrete pathlets and $\Delta' \geq 0$. Given the data structure of Lemma 5.1, we can construct in $\mathcal{O}(n \log n)$ time a target-discrete (ℓ, Δ')-pathlet $(P, \mathcal{I})$ maximizing $\|\mathrm{Cov}(P, \mathcal{I}) \setminus \mathrm{Cov}(C)\|$.*

6 Subtrajectory clustering

As stated at the end of Section 4, in the full version we give an algorithm for constructing an optimal $(2, \Delta')$-pathlet $(P, \mathcal{I})$ where P is a subsegment of an edge of S. This algorithm reports an optimal pathlet in $\mathcal{O}(n^2 \log n)$ time, using $\mathcal{O}(n)$ space. Together with the algorithm of Section 5, we can construct an optimal $(\ell, 4\Delta)$-pathlet whose reference curve is either a vertex-subcurve of S or a subsegment of an edge of S. The construction takes $\mathcal{O}(n^2 \ell \log n)$ time and uses $\mathcal{O}(n^2 \log n)$ space. The greedy set cover argument of Section 3 yields:

▶ **Theorem 6.1.** *Let T be a trajectory with n vertices, and let $\ell \in \mathbb{N}$ and $\Delta \geq 0$ be parameters. In $\mathcal{O}(kn^2 \ell \log^2 n)$ time and $\mathcal{O}(n^2 \log n)$ space, we can construct a target-discrete $(\ell, 4\Delta)$-clustering of T with at most $3k \ln n + 1$ pathlets, where k is the minimum number of pathlets in an (ℓ, Δ)-clustering.*

References

1. Pankaj K. Agarwal, Kyle Fox, Kamesh Munagala, Abhinandan Nath, Jiangwei Pan, and Erin Taylor. Subtrajectory clustering: Models and algorithms. In *proc. 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 75–87, 2018. doi:10.1145/3196959.3196972.
2. Pankaj K. Agarwal, Sariel Har-Peled, Nabil H. Mustafa, and Yusu Wang. Near-linear time approximation algorithms for curve simplification. *Algorithmica*, 42(3):203–219, 2005. doi:10.1007/s00453-005-1165-y.
3. Hugo A. Akitaya, Frederik Brüning, Erin Chambers, and Anne Driemel. Subtrajectory clustering: Finding set covers for set systems of subcurves. *Computing in Geometry and Topology*, 2(1):1:1–1:48, 2023. doi:10.57717/cgt.v2i1.7.
4. Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995. doi:10.1142/S0218195995000064.
5. Frederik Brüning, Jacobus Conradi, and Anne Driemel. Faster approximate covering of subcurves under the fréchet distance. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *proc. 30th Annual European Symposium on Algorithms (ESA)*, volume 244 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:16, Dagstuhl, Germany, 2022. doi:10.4230/LIPIcs.ESA.2022.28.
6. Kevin Buchin, Maike Buchin, David Duran, Brittany Terese Fasy, Roel Jacobs, Vera Sacristan, Rodrigo I. Silveira, Frank Staals, and Carola Wenk. Clustering trajectories for map construction. In *proc. 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10, 2017. doi:10.1145/3139958.3139964.
7. Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Jorren Hendriks, Erfan Hosseini Sereshgi, Vera Sacristán, Rodrigo I. Silveira, Jorrick Sleijster, Frank Staals, and Carola Wenk. Improved map construction using subtrajectory clustering. In *proc. 4th ACM*

SIGSPATIAL Workshop on Location-Based Recommendations, Geosocial Networks, and Geoadvertising, pages 1–4, 2020. doi:10.1145/3423334.3431451.

8 Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Maarten Löffler, and Jun Luo. Detecting commuting patterns by clustering subtrajectories. *International Journal of Computational Geometry & Applications*, 21(03):253–282, 2011. doi:10.1142/S0218195911003652.

9 Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. Approximating (k, ℓ)-center clustering for curves. In *proc. Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2922–2938, 2019.

10 Maike Buchin and Dennis Rohde. Coresets for (k, ℓ)-median clustering under the Fréchet distance. In *proc. Conference on Algorithms and Discrete Applied Mathematics*, pages 167–180, 2022.

11 Siu-Wing Cheng and Haoqiang Huang. Curve simplification and clustering under Fréchet distance. In *proc. 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1414–1432, 2023.

12 Vasek Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. doi:10.1287/MOOR.4.3.233.

13 Jacobus Conradi and Anne Driemel. Finding complex patterns in trajectory data via geometric set cover. *arXiv preprint arXiv:2308.14865*, 2023.

14 Anne Driemel, Amer Krivošija, and Christian Sohler. Clustering time series under the Fréchet distance. In *proc. twenty-seventh annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 766–785, 2016.

15 Joachim Gudmundsson and Sampson Wong. Cubic upper and lower bounds for subtrajectory clustering under the continuous Fréchet distance. In *proc. 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 173–189, 2022.

16 Leonidas J. Guibas, John Hershberger, Joseph S. B. Mitchell, and Jack Snoeyink. Approximating polygons and subdivisions with minimum link paths. *International Journal of Computational Geometry & Applications*, 3(4):383–415, 1993. doi:10.1142/S0218195993000257.

17 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *proc. symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103, 1972. doi:10.1007/978-1-4684-2001-2_9.